

embodiment is sent by a controlling device 180 to a target device 182 to cancel a pending notify command. The notify cancel command type of this alternative embodiment of the present invention has a ctype data field value of 0101. Alternatively, the notify cancel command type can have any appropriate ctype data field value. A notify cancel command type FCP data frame, according to this alternative embodiment of the present invention, is illustrated in Figure 10. The ctype data field 642 of the command FCP data frame 600 has a value of 0101 indicating that it is a notify cancel command type. This notify cancel command is sent by a controlling device 180 to a target device 182 to cancel a pending notify command. When a target device 182 receives a notify command from a controlling device 180, the target device 182 cancels the pending notify command.

A flowchart illustrating the steps within the method of the preferred embodiment of the present invention, implemented at a controlling device, is illustrated in Figure 11. The method of the present invention at the controlling device begins at the step 1000. At the step 1002, the controlling device issues a notify command to a target device. The controlling device then waits at the step 1004, until an interim response has been received. Once an interim response has been received, it is then determined at the step 1006, if the controlling device needs to cancel the pending notify command. If it is determined at the step 1006 that the controlling device does need to cancel the pending notify command, then a status command is sent to the target device at the step 1010. This status command sent while a notify command is pending at the target device, will cause the target device to cancel the pending notify command. The process then ends at the step 1012.

Otherwise, if it is determined at the step 1006 that the controlling device does not yet need to cancel the pending notify command, then it is determined at the step 1008 if a response to the pending notify command has been received. If it is determined at the step 25 1008 that a response to the pending notify command has been received, then the process ends at the step 1012. Otherwise, if it is determined at the step 1008 that a response to the pending notify command has not yet been received, then the process jumps back to the step 1006 to determine if the controlling device needs to cancel the pending notify command.

A flowchart illustrating the steps within the method of the preferred embodiment of the present invention, implemented at a target device, is illustrated in Figure 12. The method of the present invention at the target device begins at the step 1100. The target device waits at the step 1102 until a notify command is received. Once a notify command is received, the target device then sends an interim response to the controlling device at the step 1104. At the step 1106, the target device determines if its status has changed. If it is determined at the step 1106 that the status of the target device has changed, then a response to the notify command is sent to the controlling device at the step 1114. The process then ends at the step 1116.

Otherwise, if it is determined at the step 1106, that the status of the target device has not changed, then it is determined at the step 1108 if a status command has been received from the same controlling device that the notify command was received from, with the same opcode. If it is determined at the step 1108 that a status command has been received from the same controlling device with the same opcode, the target device then cancels the pending notify command at the step 1110 and sends a stable response to the status command to the controlling device at the step 1112. The process then ends at the step 1116. Otherwise, if it is determined at the step 1108 that a status command has not been received, the process jumps back to the step 1106 to determine if the status of the target device has changed.

Therefore, as described above, using the mechanism of the present invention, a controlling device has the ability to cancel a pending notify command. Preferably, the status command cancels a pending notify command only if the same operand is used between the two commands and if the status command is issued by the same controlling device that issued the original notify command. The target device determines this by inspecting the node\_ID field in the IEEE 1394 write packet and the subunit\_type and subunit\_ID fields in the AV/C frame within that same packet. Preferably, if these values in the status command match the values that were in the original pending notify command, the target device will then cancel the pending notify command. Also, preferably, a controlling device will not issue a status command to cancel a pending notify command, until the target device returns the interim response to the controlling device. This

prevents any confusion between which response should be returned first, the response to the status command or the response to the notify command. If no notify command is pending when a target device receives a status command, the target device responds to the status command as is described in the current AV/C standard.

5 The present invention has been described in terms of specific embodiments incorporating details to facilitate the understanding of the principles of construction and operation of the invention. Such references, herein, to specific embodiments and details thereof are not intended to limit the scope of the claims appended hereto. It will be apparent to those skilled in the art that modifications can be made in the embodiment chosen for illustration without departing from the spirit and scope of the invention. Specifically, it will be apparent to one of ordinary skill that while the preferred embodiment of the present invention is used with an IEEE 1394-2000 serial bus structure, the present invention could also be implemented on any other appropriate digital interfaces or bus structures, including other or later versions of the IEEE 1394 serial bus.